Highcapacity data-hiding for 3D meshes based on static arithmetic coding

Vincent Itier, William Puech, Jean-Pierre Pedeboy

HIGH CAPACITY DATA-HIDING FOR 3D MESHES BASED ON STATIC ARITHMETIC CODING

V. Itier[1,2], *W. Puech*[1], *J.-P. Pedeboy*[2]

[1]LIRMM, UMR 5506 CNRS, University of Montpellier,
860 rue de St Priest, Bat. 5, 34095 Montpellier, FRANCE
[2]STRATEGIES S.A, Parc d'Affaires Icade, Immeuble Amsterdam,
56 Rue dArcueil, 94578 Rungis, FRANCE

ABSTRACT

3D meshes are widely used today in very different domains for example; game, medical diagnostic, CAD (computed aided design) or more recently 3-D printing. In this paper, we provide a new data hiding method that has a huge capacity, $cp = 3c(n-1)$, where n is the vertex number of the mesh and c is an integer. The proposed method consists to compute a Hamiltonian path along the mesh as synchronization. At each step of path building, $3c$ bits are embedded. The embedding is designed to be a distance relation between a vertex and its father in the path. Moreover, the method uses static arithmetic coding to embed message information. We analyzed the proposed method by inserting RGB images in 3D meshes.

Index Terms— 3D data hiding, high capacity, geometry synchronization

1. INTRODUCTION

The expansion of new 3D technologies including mobile multimedia and 3D printing, requires the development of efficient 3D tools for security, adding meta-data, or integrity control. 3D mesh domain is faced with a rising tide of challenges, for instance, analysis, compression, visualization and printing. These require some security during transmission, storage, modification and sharing. A large variety of activity sectors are interested in various applications for 3D data hiding. For example, designers of 3D objects using CAD want to preserve their copyright or the integrity of the mesh. They may also want to embed meta-data about the shape changes or about its textures. The security is ensured by a secret key which remains confidential and is known only to the sender and the receiver. To ensure security, cryptography can be used for visual confidentiality, but when the media is decrypted then its security is removed as well. 3D watermarking has to enforce a series of requirements, such as the invisibility (as a function of allowed distortion), high capacity, robustness to various attacks, and to be crypto-secure as well. When enforcing any of these requirements, we are faced with a trade-off with all the others. The most important requirement is the imperceptibility, the watermarked mesh has to be the more similar to the original one. Due to this compromise, some methods focus on robustness like in [1, 2, 3], with a low payload. Other authors [4] have proposed fragile watermarking to check if the mesh has been altered since its creation. More information about 3D watermarking can be found in surveys [5, 6].

In this paper, the proposed method focuses on the embedding information capacity. We propose an iterative method, which inserts data on each vertex, one by one, as a function of the synchronization method based on a Hamiltonian path. The proposed method lacks robustness, but it offers a huge capacity. Since one of the main challenge is to not produce visible distortion, we propose to modify only the geometry of the mesh by displacing each vertex while keeping the connectivity unchanged. The synchronization, additionally giving a ordered traversal of the mesh, also provides a new relation between two vertices represented by the edges that links them in the Hamiltonian path. We choose to embed data by moving a vertex in relation to its predecessor. The new position of each vertex depends on the message and the input parameter Δ which bounds the displacement distance. To embed a part of the message, a vertex is moved in the interval Δ to a sub-interval which codes the value. By slicing the interval Δ in a chosen number of sub-intervals, the method allows us to adapt the capacity of the mesh by choosing the number of bits embedded into the vertex. The embedding method is similar to the Quantization Index Modulation (QIM) method [7]. We also propose to use redundant information from the message to provide better displacement of the vertex.

In Section 2, we first present a state of the art of 3D high capacity data hiding. The proposed method is introduced in Section 3, which is based on joint synchronization and embedding. Section 4 presents experimental results. Finally, Section 5 concludes the paper and lists our future objectives.

2. RELATED WORK

Generally, 3D processing methods and their synchronizations are defined on spatial or transformed domain which is a way to classify the techniques. Spatial domain methods are used to embed and synchronize data into the geometry of the mesh

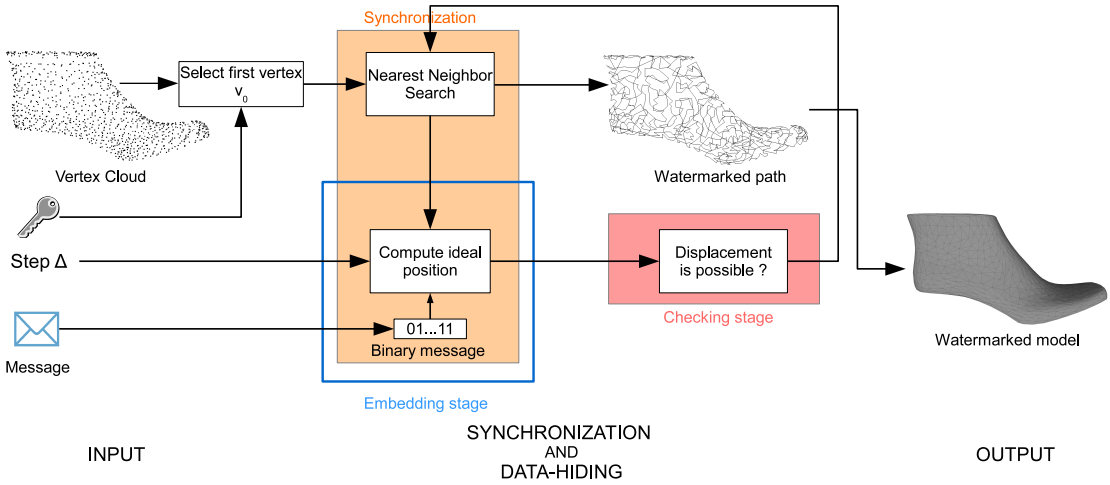


Fig. 1: Overview of the proposed embedding method.

such as the vertex position, normals, or topography. The reference high capacity methods belong to this category. Chao *et al.* [8] proposed an approach where between 21 to 31 bits can be hidden per vertex. They order vertices using topology. Then, they embed bits by displacing each vertex on x, y and z into calculated locations then they can recursively divide regions to embed more bits. Their method produces nearly invisible distortion. But the vertex scheduling stage can hinder the method robustness. Another steganography method with very high capacity, 50 bits per feature vertex, has been proposed by Li *et al.* [9]. The method adds lots of vertices which contain the payload, respecting the shape of the mesh. They also add vertices to make the mesh unsuspicious but then its size increases which can be detected. For robustness purpose, Gao *et al.* [10] proposed a semi-blind watermarking which is based on affine-invariants. Watermark codes are inserted by modifying the length ratio of each vertex. Their method can withstand the majority of classical attacks, and the capacity depends on the range which is proportional to distortion. According to their parameters, their capacity can be easily between 1 to 2 bits per vertex. Nevertheless, they have to assign an unique index to each piece of the message to resynchronize it. This synchronization issue prevents to embed more bits. Bogomjakov *et al.* [11] method has a high embedding capacity computed as $\lfloor \sum_{i=1}^{n} log_2 i \rfloor + \lfloor \sum_{i=1}^{m} log_2 i \rfloor$, where n is the number of vertices and m the number of facets of the mesh. Their method is fast and the data hiding is done by changing the order of vertices in the file. This method is interesting because it does not modify the 3D mesh, thus the message is not included in the mesh but in the file itself. However, this method is not robust to mesh file reshuffling.

3. PROPOSED METHOD

In Section 3.1 we give an overview of the whole method. We define the synchronization in Section 3.2 and we present the method to embed data in Section 3.3. Finally in Section 3.4, we deal with the causality issue to keep the synchronization.

3.1. Overview

Fig. 1 illustrates the overview of the proposed method, in which we use two main steps which are the synchronization and the embedding. In Fig. 1, we present the input of the method i.e. a 3D vertex cloud, the message and two parameters, the key which gives the first vertex and Δ a displacement bound. The method is based on the conjunction of synchronization and embedding. As the synchronization is very sensitive to vertex displacement, we propose to jointly synchronize and embed data, vertex by vertex in the building path order. The synchronization analyzed in Section 3.2, consists of finding the nearest neighbor of the current vertex. Then, we can compute the ideal position explained in Section 3.3, of the new current vertex as a function of is father, the message and Δ.

This iterative process allows to build the subpath $\mathbf{P_i}$ at the step $i < n$, without taking into account the next chosen vertex. Indeed, we want to perform the same order along the vertex graph at the decoding stage. To be sure to keep this exact same order, we propose a checking stage where we verify if a displacement is allowed. The result of the proposed method is a 3D watermarked mesh which has the lowest distortion possible.

3.2. Synchronization

The synchronization of the proposed method is designed to order the vertices of a mesh in a unique way. For a mesh $\mathcal{M} = (V_n, E_j)$, where V_n is the vertices set and E_j is the edges set, the Hamiltonian path is built over the complete graph of the vertices with the aim to not depend on the connectivity E_j. The weighted graph is denoted $G_n = (V_n, E_m, w)$ and is defined as the complete graph on all vertices of the mesh, where w: $E \rightarrow \mathbb{R}^+$. E_m represents the edges of the graph with $m = n(n-1)/2$. Note that the topology of the mesh $E_j \in E_m$. Each edge e_i has a weight defined as its length in Euclidean distance. The path constructed at the step i is a

sub-Hamiltonian path $\mathbf{P_i}$ on the set $V_i = \{v_0, \ldots, v_i\}$, which is made up of the chosen edges $\{e_0, \ldots e_{i-1}\}$. Any vertex is either in V_i, or in $V_n \backslash V_i = \{v_{i+1}, \ldots, v_n\}$ the set of the non-visited vertices. We use this secret key to choose the first vertex v_0 over the n vertices to build a Hamiltonian path which is a path that visits each vertex exactly once. In order to build it, we recursively choose the nearest neighbor v_{i+1} of the current vertex v_i, $\in V_i$ by choosing the minimal edge $e_i \in E_m$, such as $w(e_i) = min\|v_i, v_k\|_2, v_k \in V_n \backslash V_i$. Obviously, after the last step, $\mathbf{P_n}$ is a Hamiltonian path on G_n.

3.3. Embedding data

In Section 3.2, we create new relations between two vertices, given by the edges of the path $\mathbf{P_n}$. We use the size of new edges to define the displacement of a vertex. For each iteration $i, 0 \leq i < n$, of the construction of the path, we move the vertex v_{i+1} relatively to its predecessor v_i. Moreover, to have a larger embedding domain, we propose to displace the vertex on the three parameters ρ, θ and ϕ after conversion change the Cartesian coordinates to spherical ones. The main idea is to define a mobility interval Δ where each vertex component c_{i+1} is moved to its new value c'_{i+1}. In order to find the lower boundary of the interval b_l, we calculate:

$$b_l = \lfloor \frac{c_{i+1}}{\Delta} \rfloor \times \Delta. \quad (1)$$

The upper boundary is: $b_u = b_l + \Delta$. Thus, for each coordinate, we define a range for the new value. The value of Δ is discussed in Section 4, but it has to be small in order not to produce visible distortions. Furthermore, for embedding data, we propose to split this interval, in sub-intervals corresponding to values of the message $\mathbf{M}$. As we are limited by the numerical precision, we split the interval up to 256 sub-intervals denoted as $\delta_i, i \in [0, 256]$, i.e. we can embed up to 8 bits into a single coordinate.

To split the interval Δ, in x sub-intervals, $x = 2^k, k \in [0, 8]$, we propose to use the information contained in the message. It means that we use the probability distribution to set the size of each sub-interval. The method relies on static arithmetic coding (SAC) which was introduced by Langdon [12]. The principle of the SAC is to represent a symbols sequence by an interval of real number between 0 and 1. Each value, in this interval, corresponds to an unique word to be coded. The SAC starts to computes the probabilities, then it associates to each symbol the corresponding sub-interval. Indeed, we know exactly what letters s_j are used, and in what proportion over an alphabet $\mathcal{S}$ which has q symbols, $q \leq x$. Therefore, the probability of each value $s_j, j \in [0, q]$ is given by its membership probability in the message $\mathbf{M}$ given by $p(s_j) = \frac{|\mathbf{M}|_{s_j}}{|\mathbf{M}|}$. Each sub-interval δ_{s_j}, has a size which depends on the probability $p(s_j)$, $|\delta_{s_j}| = \frac{p(s_j)}{\Delta}$.

Note that, as we want to keep our method blind, we have to insert the probabilities distribution in the beginning of the message. This would affect the capacity because we use the same number of vertices as the size of $\mathcal{S}$. Compared to the capacity of the method, the loss is not significant. Regarding the embedding, we have just to associate each word to its corresponding sub-interval. But to reduce distortions, instead of fixing the new coordinate value to the middle of the sub-interval, we choose to fix it to the lower boundary or close to the upper boundary. The choice of the location depends on the previous coordinate value:

$$c'_{i+1} = \begin{cases} b_l + w_b & \text{if } c_{i+1} < b_l, \\ b_l + w_{b+1} - \gamma & \text{else} \end{cases} \quad (2)$$

where γ is $\frac{1}{k}$ of the sub-interval size $k \in \mathcal{N}, k > 1$.

3.4. Causality issue

The causality issue can occur when we move a vertex. It means that embedding data causes the loss of the synchronization. To prevent it, we propose a checking stage, illustrated in the overview from Fig. 1, that permits a displacement. Thus, there are two cases: modifying the path that we have been building, and the loss of the new edge that is currently created. In first case, the vertex v_{i+1} could be moved too close from the sub-path $\mathbf{P_i}$ and a vertex of the sub-path $v_j \in \mathbf{P_i}, j \neq i$ could become its predecessor. In the second case, the vertex v_{i+1} is moved to a new position v'_{i+1} which is too far from it predecessor and another vertex $v_j, \in V_n \backslash V_i$ becomes nearest to v_i. To deal with this issue, we do not move a vertex if its new location falls in a situation described above.

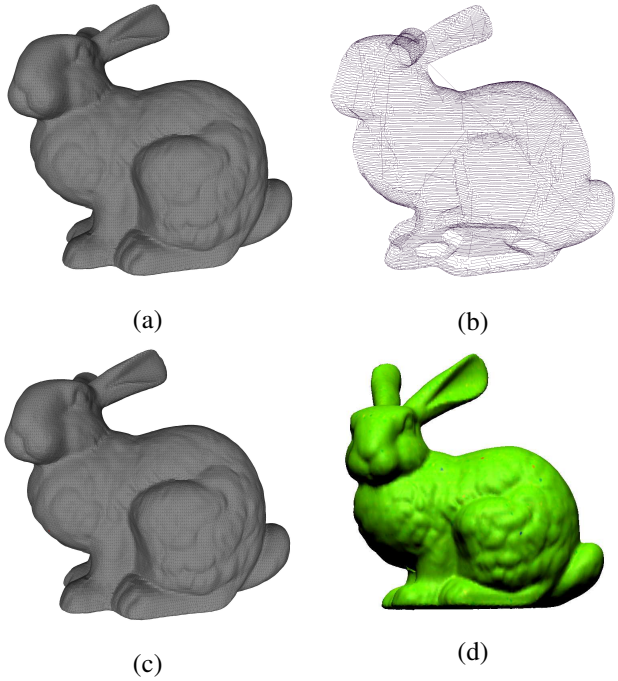
4. EXPERIMENTAL RESULTS

We present an example of the proposed method for the bunny model which has 34834 vertices, illustrated in Fig. 2.a. The model is normalized by setting the mean of edge weights to 1. We use a value of $\Delta = 10^{-4}$ that produces invisible distortion and comparable results with other techniques, we also fix the value of $\gamma = \frac{1}{10}$ Eq. (2). In this 3D object, we can insert 836016 bits and we want to insert a square RGB image. We use the proposed method by inserting a pixel per vertex, i.e. 8 bits per coordinate, which corresponds to a color channel. We resize the image to 185×185 which corresponds to $185 \times 185 \times 3 \times 8 = 821400$ bits. And we have to embed the distribution of each color, in the first vertices i.e. $256 \times 3 = 768$ vertices. Table 1 presents the differences between the original message Fig. 3.a. and the extracted message Fig. 3.b. for each method in terms of PSNR and BER (bit error rate).

Table 1: Results for the message embedding Bunny model.

	Uniform method	SAC method
# of no moved vertex	5	6
PSNR (dB)	39,8238	39,2722
BER ($\times 10^{-6}$)	134,4	163,6

We note that in order to avoid causality effects, we have lost a very small percentage of information. Nevertheless,



(a) (b) (c) (d)

Fig. 2: a) Original mesh with 34834 vertices, c) Hamiltonian path on the vertex cloud, d) Watermarked mesh, e) Distortion comparison.

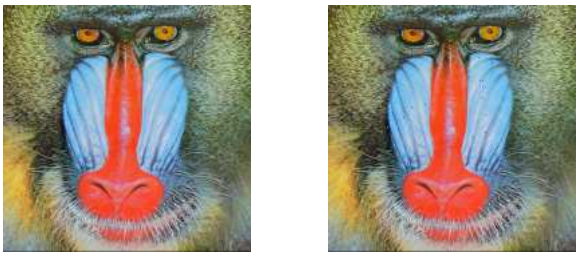
the extracted image has a good quality. We want to stress that we could have repaired false pixels by inserting an error correcting code. In order to compare our method with the previous ones, Table 2 presents the results of the methods in terms of capacity, Hausdorff [13] distance and $PSNR_1$ [8] to validate the quality of the watermarked mesh, Fig. 2.c.

Table 2: Comparison with previous methods on Bunny model.

	Capacity (bits)	H Distance	$PSNR_1$
Chao *et al.* [8]	940464	$\times$	100.57
Gao *et al.* [10]	51408	$0,548.10^{-3}$	70.02
Ours	821400	$1,024.10^{-3}$	127.236

Results show that our method has a better trade-off between the capacity and the imperceptibility. Fig. 2.d. illustrates the difference between the two meshes, small distances are represented in green, larger negative distances tend to blue, and larger positive distances tend to red. To assess this impression, we have produced results on a large database composed of 31 3D models[1], that provides a representative diversity of shapes and sizes. The number of vertices of 3D meshes is in between 1000 and 199093. Furthermore, we use the same normalization and parameters as in the full example. We use two distortion metrics, the $PSNR_1$ [8] and the $MSDM2$ [14] which is more correlated with human perception. $MSDM2$ distance has a value between 0 and 1, where 0 is for no distortion and close to 1 for huge degradations. In order to evaluate the extracted message quality, we propose to use the PSNR and the bit error rate (BER) to compare bits flows. The mean of results is presented in Table. 3.

[1]Strategies S.A, MADRAS project, Stanford University, LGMA



(a) (b)

Fig. 3: a) Original message, b) Extracted message.

Table 3: Mean results in terms of mesh distortion and message quality, for the SAC method obtained on 31 objects.

capacity	mesh distortion		message quality	
	$PSNR_1$	$MSDM2$	PSNR	BER
38421	125,459	0,003196	34,99683	0,051943

The distortion, in terms of MSDM2, corresponds to almost invisible distortion for HSV. We note that some meshes have a bad quality of message. Such meshes have a majority of the vertices that cannot be moved and consequently they fail at the checking stage, because their mesh is very regular. However, in general the proposed method has a good trade-off between imperceptibility, capacity, and quality of the message.

5. CONCLUSION

In this paper, we have proposed a blind high capacity data hiding method which is designed for embedding a color image in graphical objects. However, the scheme of the method can be used to insert any kind of data. We use a new synchronization based on the geometry of the mesh which orders the vertices by building a Hamiltonian path from a starting vertex. The synchronization algorithm is improved in order to jointly embed the data to be hidden. The message is embedded by slightly moving a vertex from its predecessor in the path. Results show that the proposed method has a high capacity 3 to 24 bits per vertex and produced non visible changes.

We list possible ideas for future exploration. The first improvements will be to compute the step Δ reliably and automatically, that could be useful for a non expert user, but it also enforces Δ to be secret from a potential attacker. Another improvement is to make the method more secure by using the secret key in the path building to generate a non deterministic path. On the other hand, we are analyzing the causality effect to find a more efficient strategy to deal with it and to improve the robustness of the synchronization.

6. REFERENCES

[1] K. Wang, G. Lavoué, F. Denis, and A. Baskurt, "Robust and blind mesh watermarking based on volume moments," *Computer Graphics*, vol. 35, no. 1, pp. 1–19, 2011.

[2] A. G. Bors and M Luo, "Optimized 3D watermarking for minimal surface distortion," *IEEE Transactions on Image Processing*, vol. 22, no. 5, pp. 1822–1835, 2013.

[3] X. Rolland-Neviere, G. Doerr, and P. Alliez, "Triangle surface mesh watermarking based on a constrained optimization framework," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 9, pp. 1491–1501, Sept 2014.

[4] B.-L. Yeo and M.M. Yeung, "Watermarking 3D objects for verification," *IEEE Computer Graphics and Applications*, vol. 19, no. 1, pp. 36–45, 1999.

[5] P. Rondao Alface and B. Macq, "From 3D mesh data hiding to 3D shape blind and robust watermarking: A survey," in *Transactions on Data Hiding and Multimedia Security II*, YunQ. Shi, Ed., vol. 4499, pp. 91–115. Springer Berlin Heidelberg, 2007.

[6] K. Wang, G. Lavoué, F. Denis, and A. Baskurt, "A comprehensive survey on three-dimensional mesh watermarking," *IEEE Transactions on Multimedia*, vol. 10, no. 8, pp. 1513–1527, 2008.

[7] B. Chen and G. W. Wornell, "Quantization index modulation methods for digital watermarking and information embedding of multimedia," *Journal of VLSI signal processing systems for signal, image and video technology*, vol. 27, no. 1/2, pp. 7–33, 2001.

[8] M.-W. Chao, C.-H. Lin, C.-W. Yu, and T.-Y. Lee, "A high capacity 3D steganography algorithm," *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, no. 2, pp. 274–284, 2009.

[9] M. T. Li, N. C. Huang, and C. M. Wang, "A novel high capacity 3D steganographic algorithm," in *International Journal of Innovative Computing, Information and Control*, 2011, vol. 7, pp. 1055–1074.

[10] X. Gao, C. Zhang, Y. Huang, and Z. Deng, "A robust high-capacity affine-transformation-invariant scheme for watermarking 3D geometric models," *ACM Transaction on Multimedia Computing, Communications and Applications*, vol. 8, no. 2S, pp. 34:1–34:21, 2012.

[11] A. Bogomjakov, C. Gotsman, and M. Isenburg, "Distortion-free steganography for polygonal meshes," in *Computer Graphics Forum*, 2008, vol. 27, pp. 637–642.

[12] G. G. Langdon, "Arithmetic coding.," *IBM Journal of Research and Development*, vol. 23, pp. 149–162, 1979.

[13] P. Cignoni, C. Rocchini, and R. Scopigno, "Metro: Measuring error on simplified surfaces," Tech. Rep., Centre National de la Recherche Scientifique, 1996.

[14] G. Lavoué, "A multiscale metric for 3D mesh visual quality assessment," *Computer Graphics Forum*, vol. 30, no. 5, pp. 1427–1437, 2011.